

Image Registration

Lecture 8: Image Resampling

Prof. Charlene Tsai

Why Resampling ?

**Resampling
is the Essence
of Intensity Based
Image Registration**

What is an Image ?

An Image is
a sampling of a
continuous field
using a discrete grid

Image Origin & Spacing

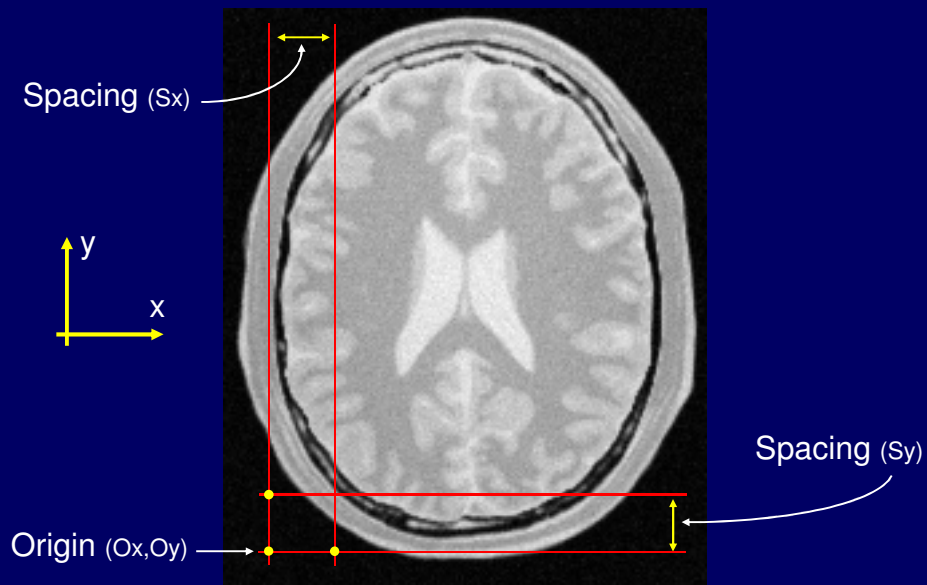


Image Sampling Grid

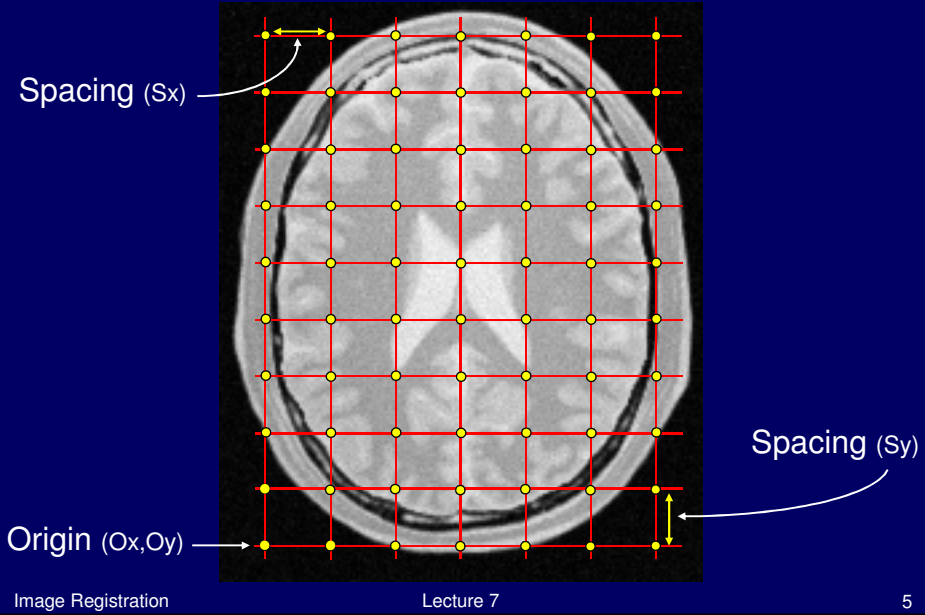


Image Pixel

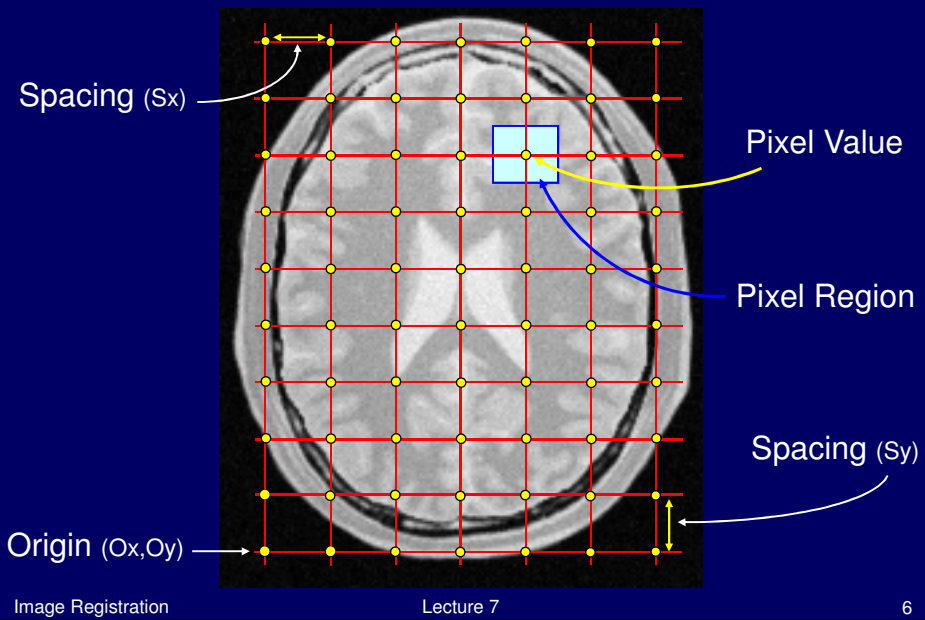
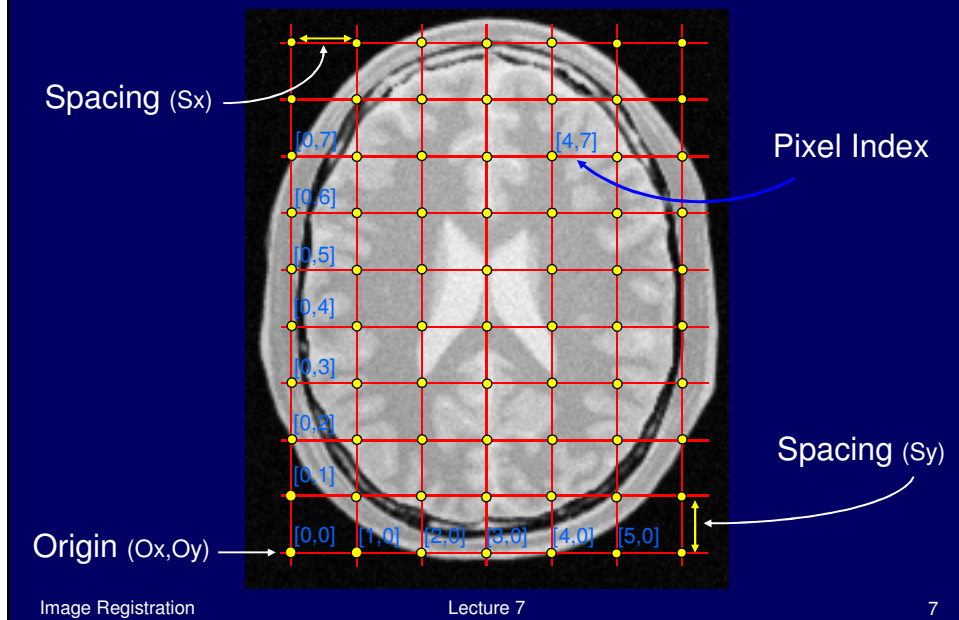
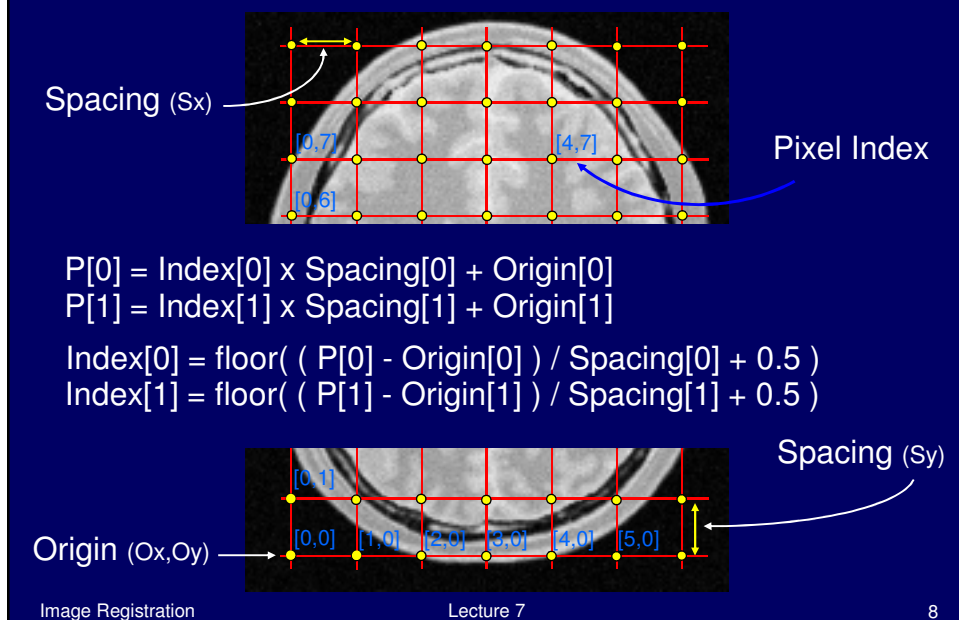
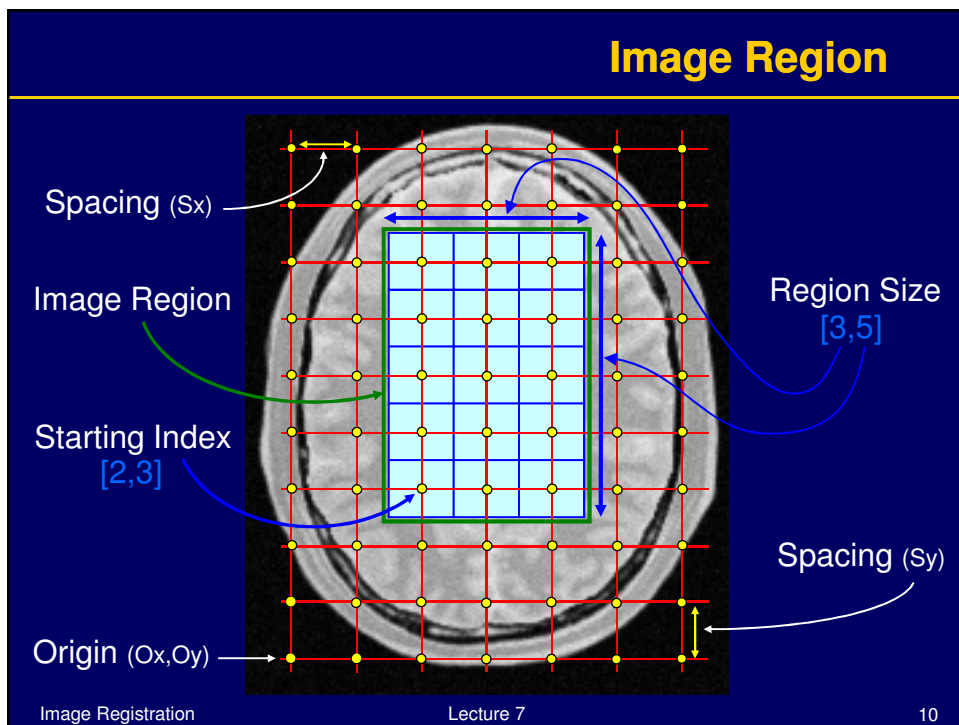
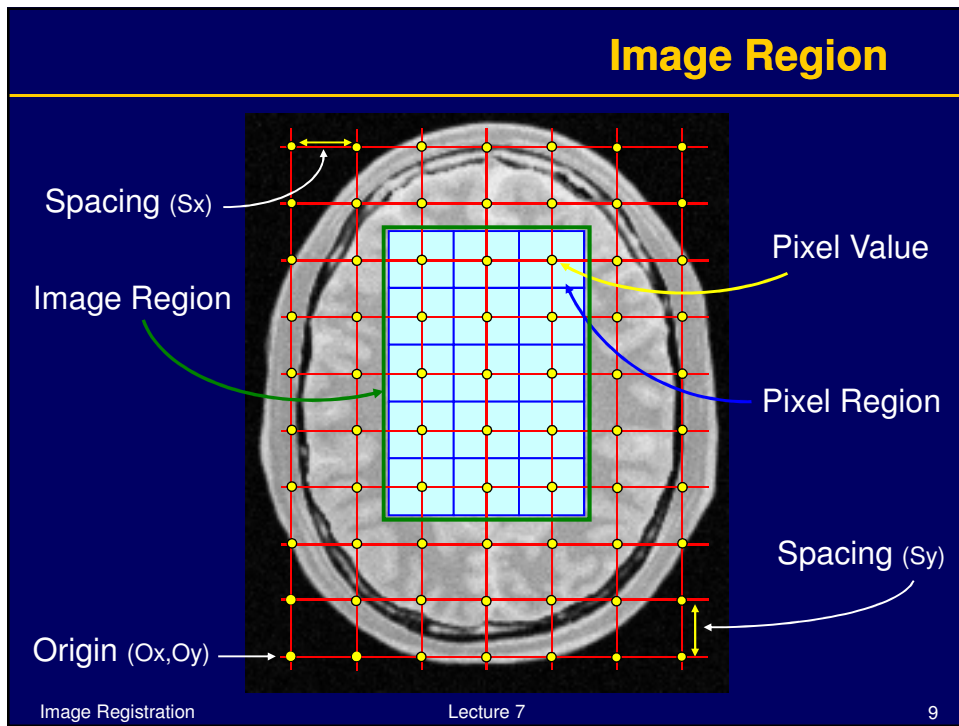


Image Indices



Index to Physical Coordinates

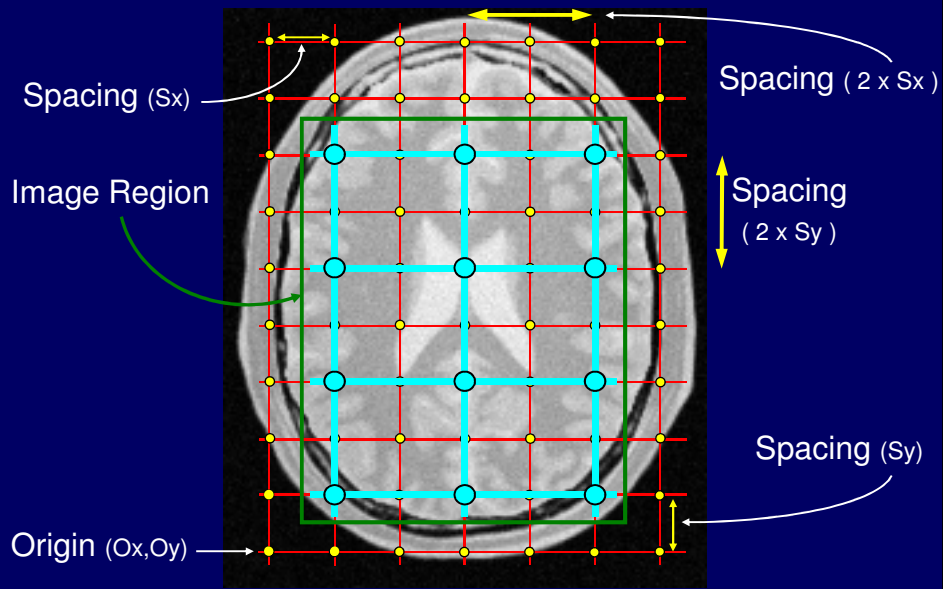




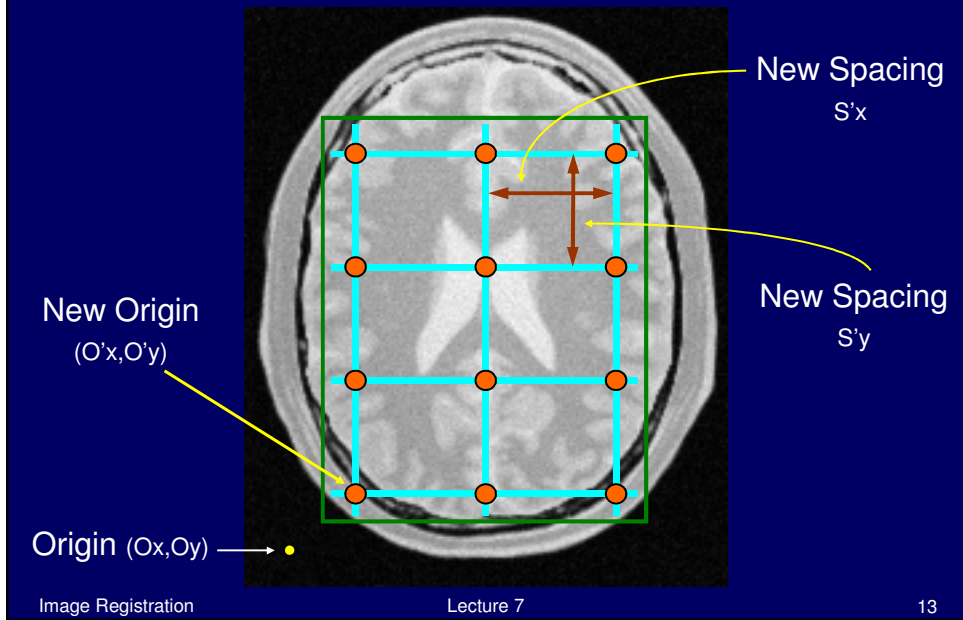
Basic Resampling

Resampling Trivial Cases

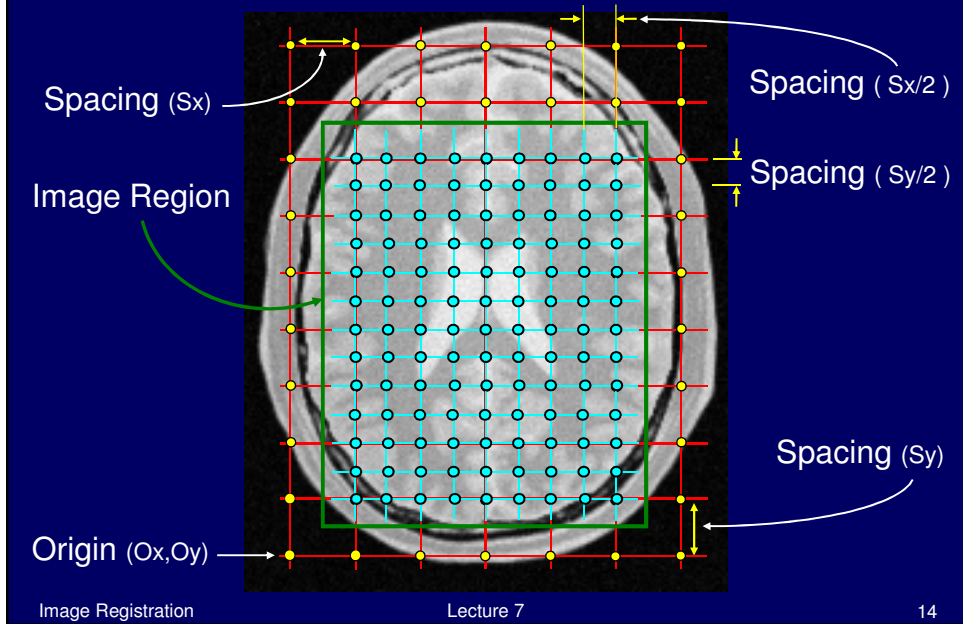
Sub-Sampling by Half



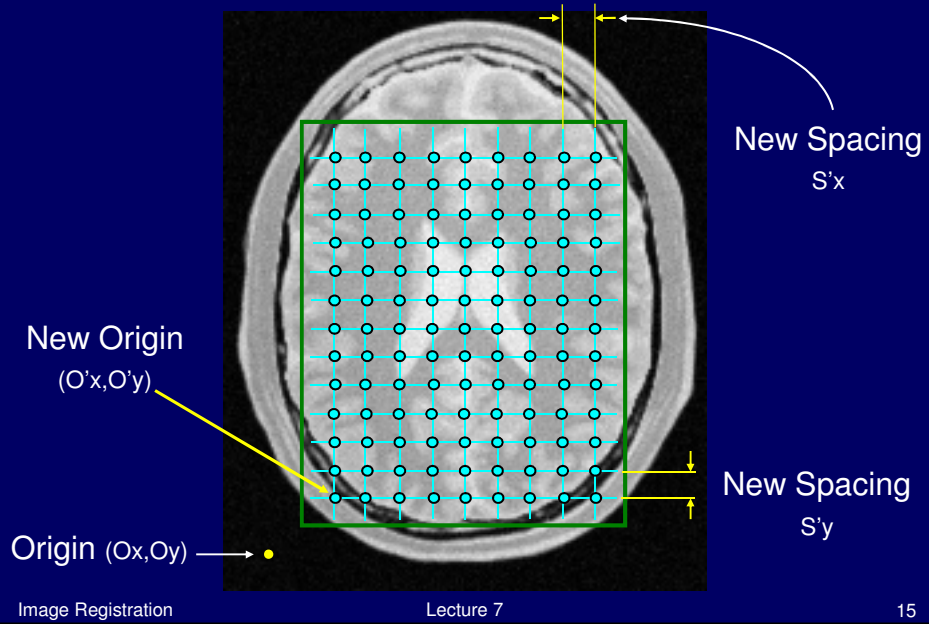
Sub-Sampling by Half



Super-Sampling by Double



Super-Sampling by Double



Resampling in ITK

`itk::ResampleImageFilter`

Resampling in ITK

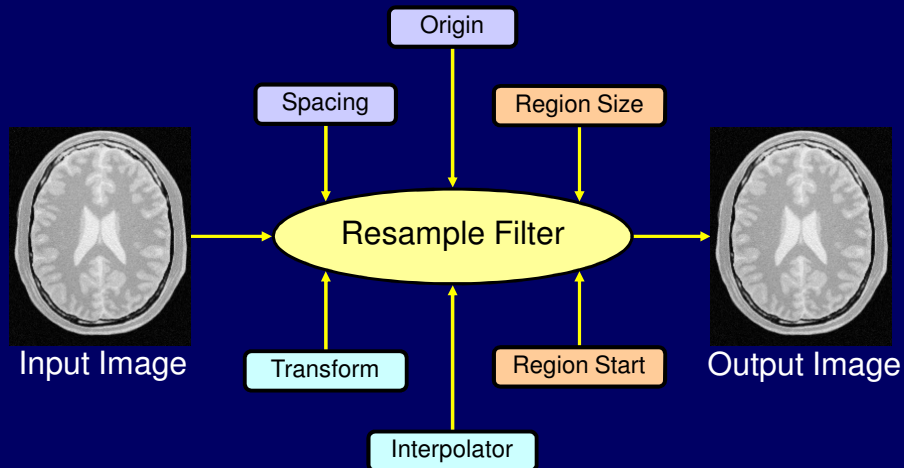


Image Registration

Lecture 7

17

Resample Image Filter

```
#include "itkImage.h"
#include "itkResampleImageFilter.h"
#include "itkIdentityTransform.h"
#include "itkLinearInterpolateImageFunction.h"

typedef itk::Image< char, 2 > ImageType;

ImageType::Pointer inputImage = GetImageSomeHow();

typedef itk::ResampleImageFilter< ImageType > FilterType;

FilterType::Pointer resampler = FilterType::New();

ImageType::SizeType size;
size[0] = 200;
size[1] = 300;

ImageType::IndexType start;
start[0] = 0;
start[1] = 0;
```

Image Registration

Lecture 7

18

Resample Image Filter

```
ImageType::PointType origin;
origin[0] = 10.0; // millimeters
origin[1] = 25.5; // millimeters

ImageType::SpacingType spacing;
spacing[0] = 2.0; // millimeters
spacing[1] = 1.5; // millimeters

resampler->SetOutputSpacing( spacing );
resampler->SetOutputOrigin( origin );

resampler->SetSize( size );
resampler->SetOutputStartIndex( start );

resampler->SetDefaultPixelValue( 100 );

resampler->SetInput( inputImage );
```

Resample Image Filter

```
typedef itk::LinearInterpolateImageFunction<
    ImageType,
    double > InterpolatorType;

InterpolatorType::Pointer interpolator = InterpolatorType::New();

typedef itk::TranslationTransform< double, 2 > TransformType;

TransformType::Pointer transform = TransformType::New();

transform->SetIdentity();

resampler->SetInterpolator( interpolator );
resampler->SetTransform( transform );

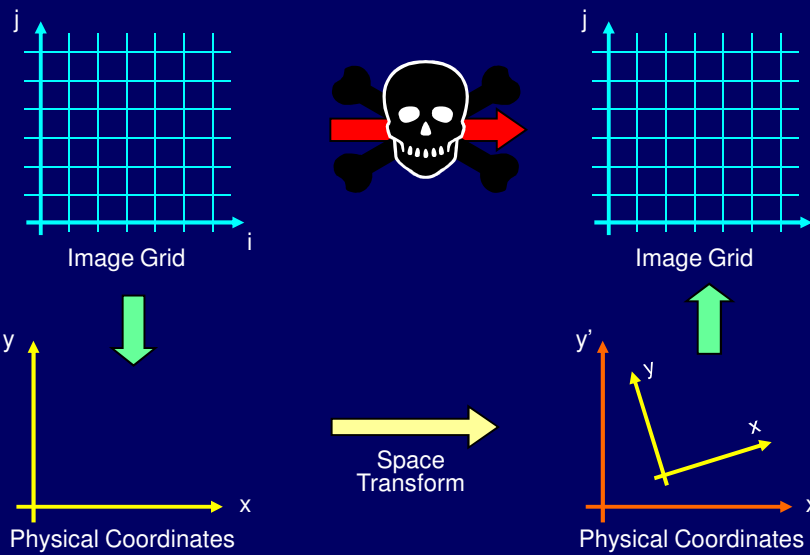
resampler->Update();

const ImageType * outputImage = resampler->GetOutput();
```

Basic Registration

Registration Basics

Coordinate System Conversions

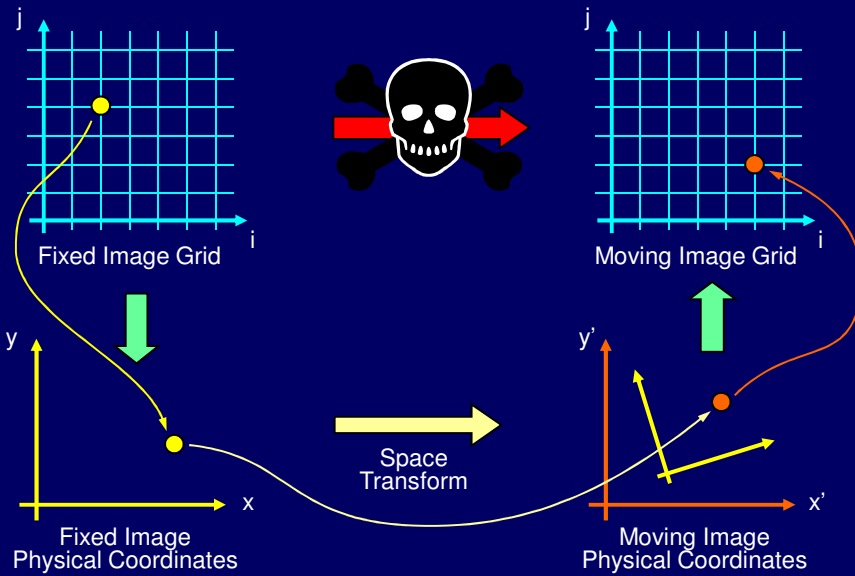


Things I will not do...

*I will not register images in pixel space
I will not register images in pixel space
I will not register images in pixel space
I will not register images in pixel space
I will not register images in pixel space
I will not register images in pix*



Fixed Image & Moving Image



Selecting Moving & Fixed Images

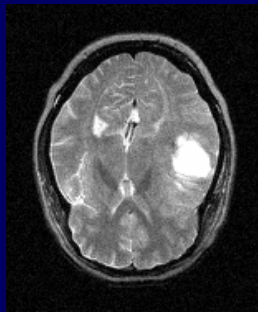
In principle the denomination of
Fixed Image & Moving Image
is arbitrary

In practice the moving image
is the one that will be resampled
into the fixed image
coordinate system

Quiz #1

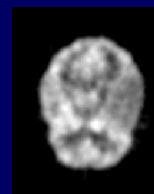
Images from the same patient

MRI-T2



256 x 256 pixels

PET



128 x 128 pixels

Moving Image ?

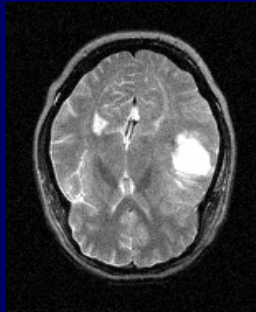
Fixed Image ?

Images provided as part of the project: "Retrospective Image Registration Evaluation",
NIH, Project No. 8R01EB002124-03, Principal Investigator, J. Michael Fitzpatrick, Vanderbilt University, Nashville, TN.

Quiz #2

Images from the same patient

MRI-T2



256 x 256 pixels

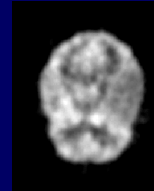


Scaling Transform

What scale factor ?

- a) 2.0
- b) 1.0
- c) 0.5

PET



128 x 128 pixels

Images provided as part of the project: "Retrospective Image Registration Evaluation",
NIH, Project No. 8R01EB002124-03, Principal Investigator, J. Michael Fitzpatrick, Vanderbilt University, Nashville, TN.

Security

Airport Security

"Do not accept packages
or baggage from unknown individuals."

Image Security

"Do not accept images
or volumes without pixel spacing."

Exercise #1

Input Image

Size: 181x217 pixels

Spacing: 0.8 x 0.9 mm

Output Image

Size: 181x217 pixels

Spacing: 0.8 x 0.9 mm

Coordinates ?
(0.0mm , 195.3mm)

Coordinates ?
(144.8mm , 195.3mm)

Output Origin
(35.0mm , 50.0mm)

Origin: (0,0)

Coordinates ?
(144.8mm , 0.0mm)

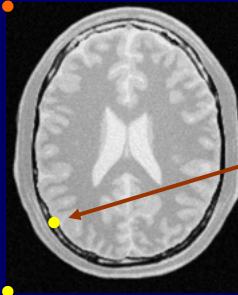


Image Registration

Lecture 7

29

Exercise #1

Input Image

Size: 181x217 pixels

Spacing: 0.8 x 0.9 mm

Output Image

Size: 181x217 pixels

Spacing: 0.8 x 0.9 mm

Origin: (0,0)

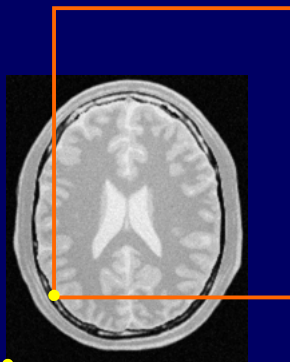


Image Registration

Lecture 7

30

Resample Image Filter

```
#include "itkImage.h"
#include "itkResampleImageFilter.h"
#include "itkIdentityTransform.h"
#include "itkNearestNeighborInterpolateImageFunction.h"

typedef itk::Image< char, 2 > ImageType;

ImageType::ConstPointer inputImage = reader->GetOutput();

typedef itk::ResampleImageFilter< ImageType > FilterType;

FilterType::Pointer resampler = FilterType::New();

ImageType::RegionType region = inputImage->GetBufferedRegion();

resampler->SetSize( region->GetSize() );
resampler->SetOutputStartIndex( region->GetIndex() );

resampler->SetOutputSpacing( inputImage->GetSpacing() );
```

Resample Image Filter

```
ImageType::PointType origin;
origin[0] = 35.0; // millimeters
origin[1] = 50.0; // millimeters

resampler->SetOutputOrigin( origin );

typedef itk::IdentityTransform< double, 2 > TransformType;

TransformType::Pointer transform = TransformType::New();

resampler->SetTransform( transform );

resampler->SetDefaultPixelValue( 100 );

resampler->SetInput( inputImage );

writer->SetInput( resampler->GetOutput() );

writer->Update();
```


Exercise #1

Input Image

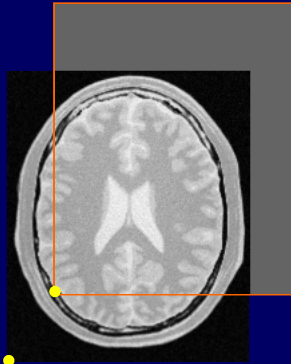
Size: 181x217 pixels

Spacing: 0.8 x 0.9 mm

Output Image

Size: 181x217 pixels

Spacing: 0.8 x 0.9 mm



Origin: (0,0) →

Image Registration

Lecture 7

33

Quiz #3

Fixed Image

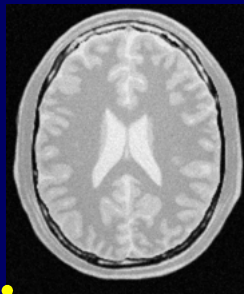
Size: 181x217 pixels

Spacing: 0.8 x 0.9 mm

Moving Image

Size: 181x217 pixels

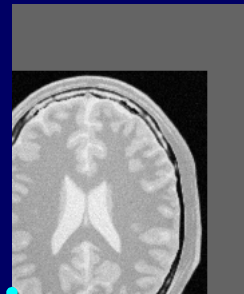
Spacing: 0.8 x 0.9 mm



Origin: (0,0)



Transform ?



Origin: (35,50)

Image Registration

Lecture 7

34

Things I will not do...

*I will not register images in pixel space
I will not register images in pixel space
I will not register images in pixel space
I will not register images in pixel space
I will not register images in pixel space
I will not register images in pixel space*

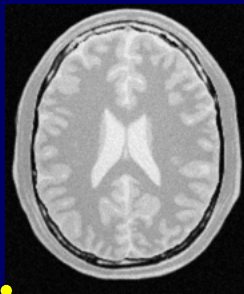


Quiz #4

Fixed Image

Size: 181x217 pixels

Spacing: 0.8 x 0.9 mm

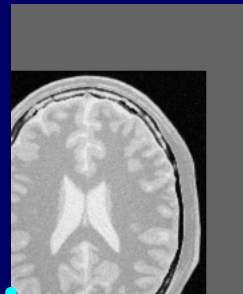


Origin: (0,0)

Moving Image

Size: 181x217 pixels

Spacing: 0.8 x 0.9 mm



Origin: (0,0)



Transform ?

Return to the Source

Input Image

Size: 181x217 pixels

Spacing: 0.8 x 0.9 mm

Output Image

Size: 181x217 pixels

Spacing: 0.8 x 0.9 mm

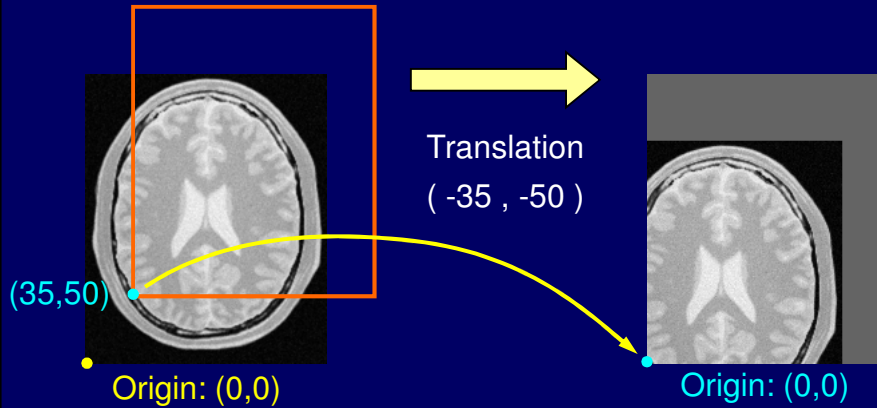


Image Registration

Lecture 7

37

More Resampling

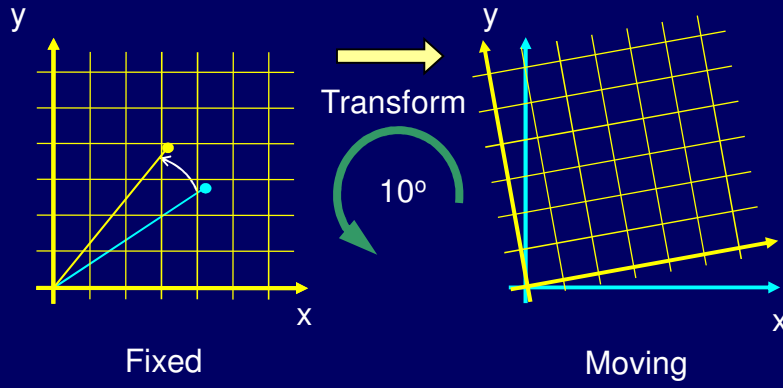
Resampling
Less Trivial Cases

Image Registration

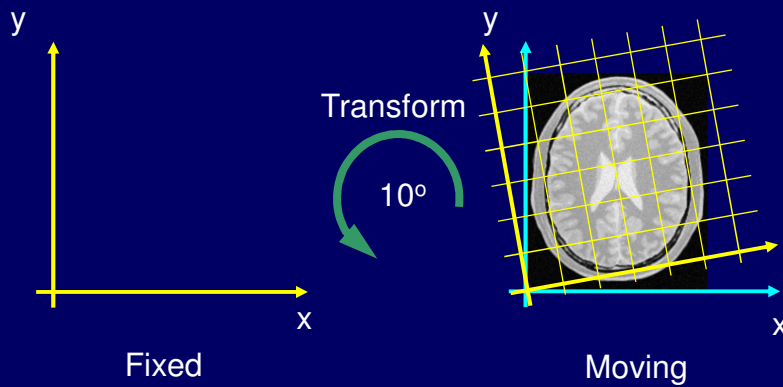
Lecture 7

38

Rotation



Rotation



Rotation

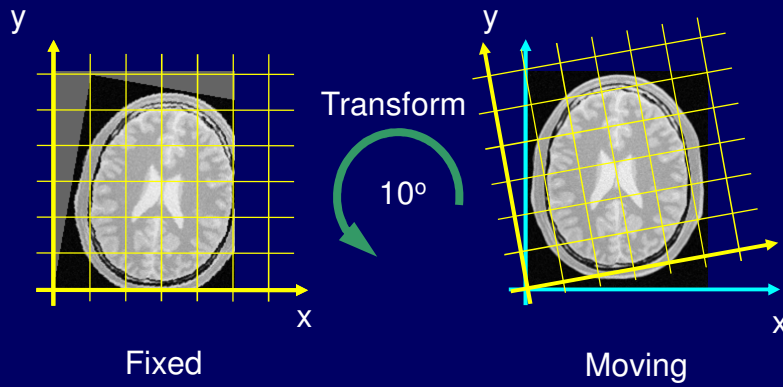


Image Registration

Lecture 7

41

Rotation around the Origin

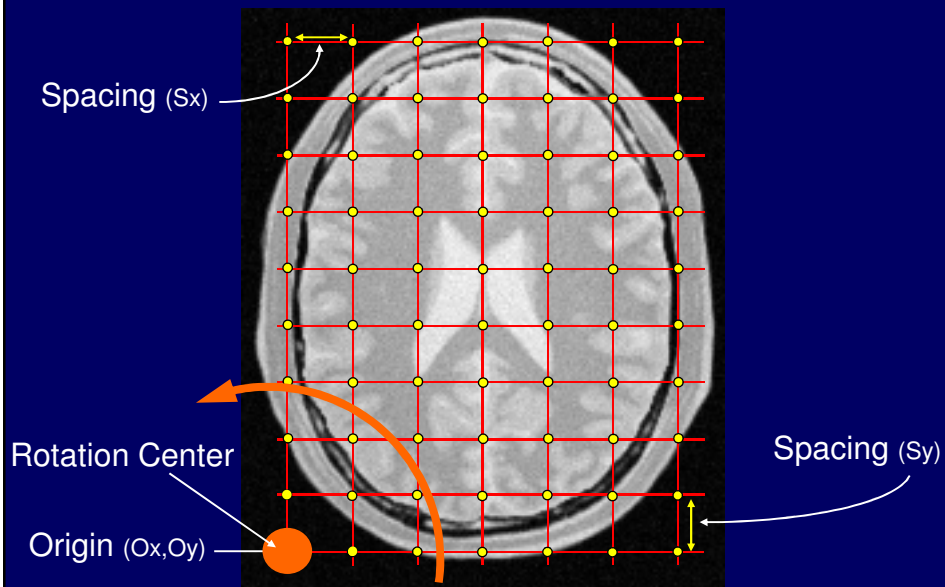
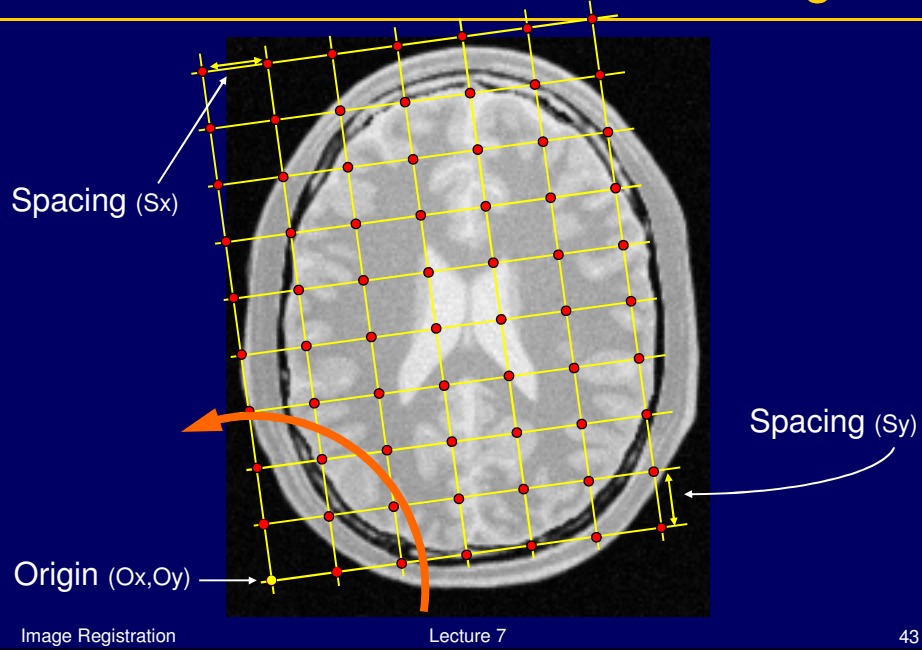


Image Registration

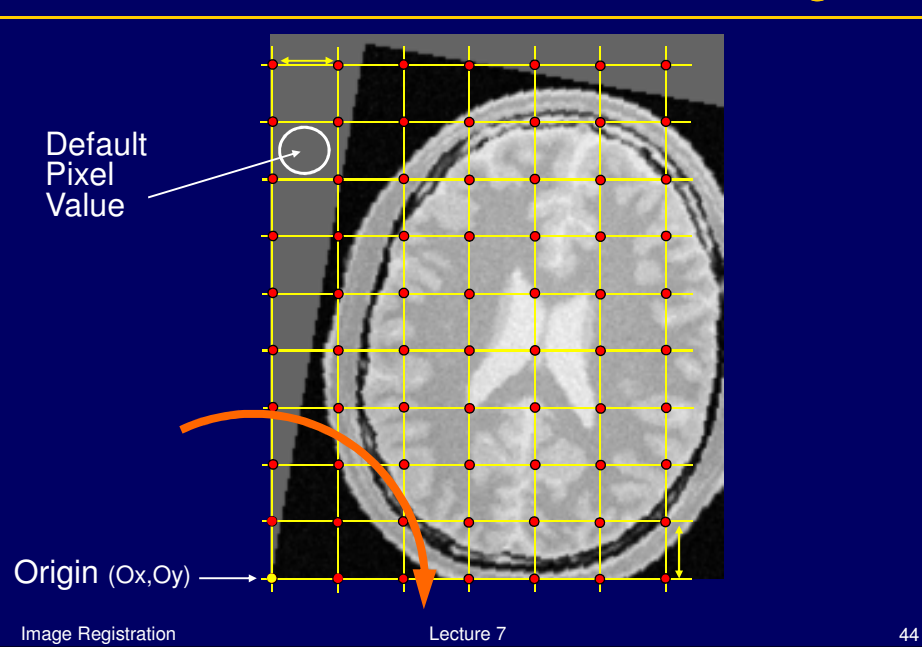
Lecture 7

42

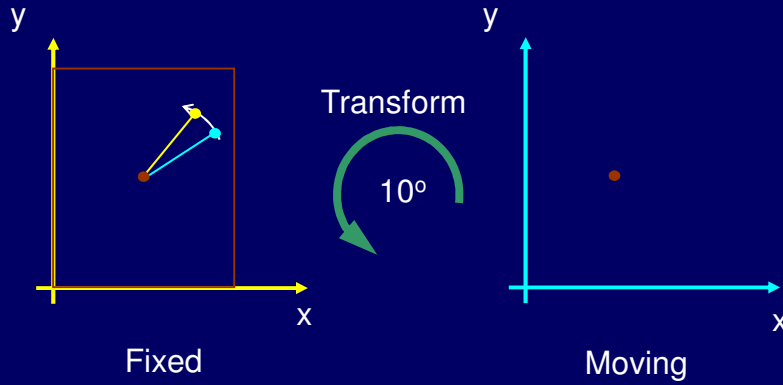
Rotation around the Origin



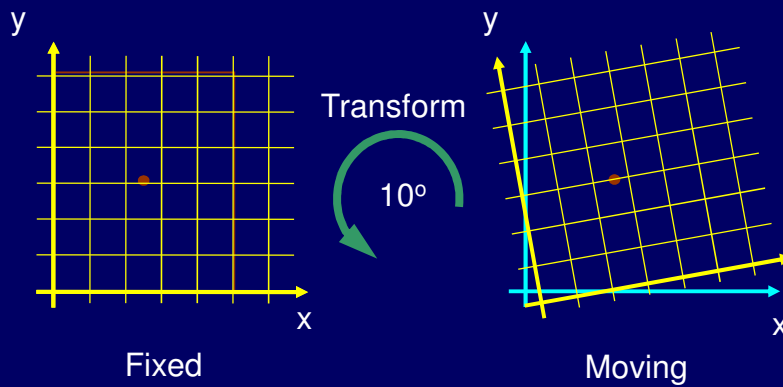
Rotation around the Origin



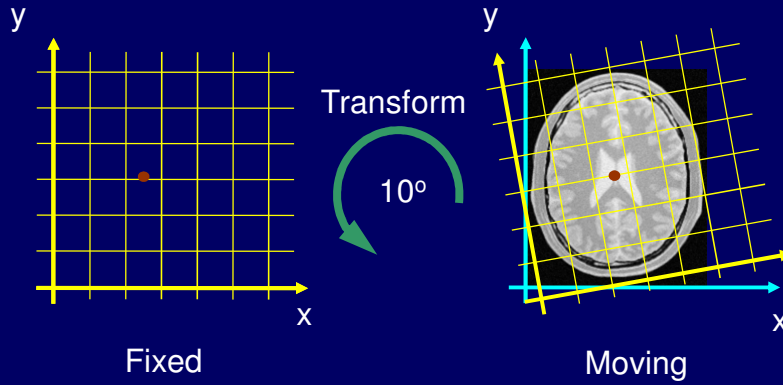
Rotation around the Center



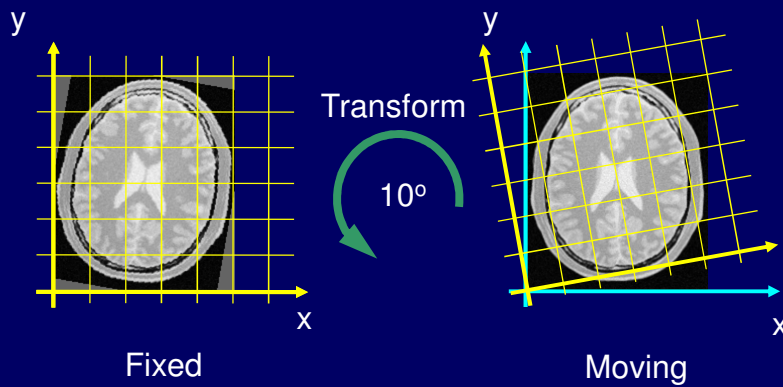
Rotation around the Center



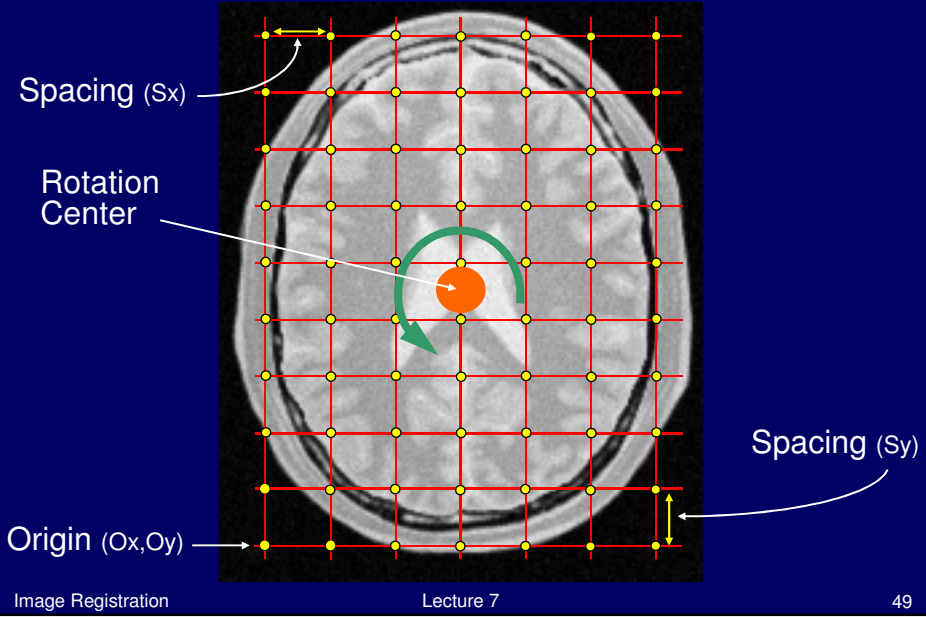
Rotation around the Center



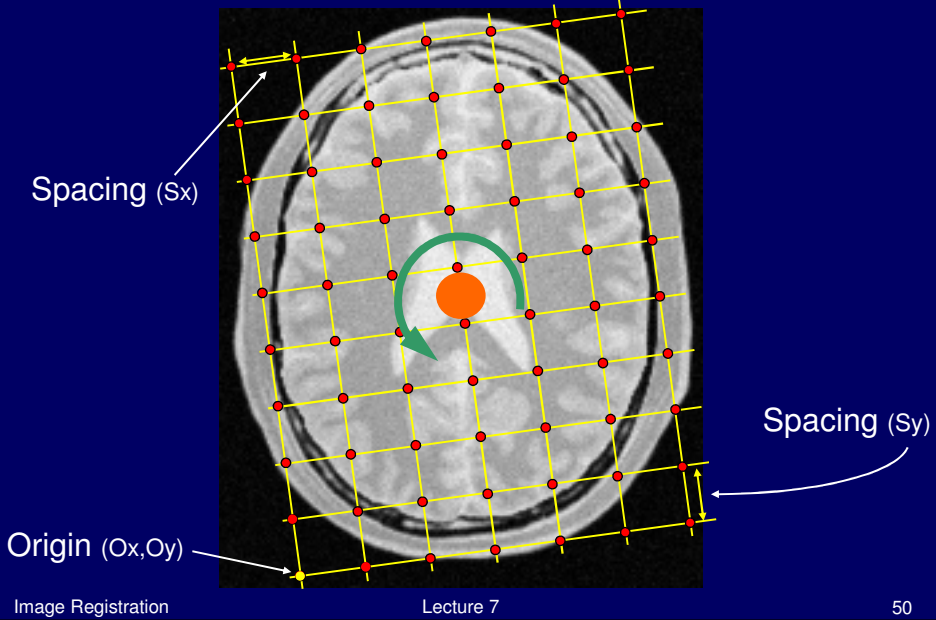
Rotation around the Center



Rotation around the Center



Rotation around the Center



Rotation around the Center

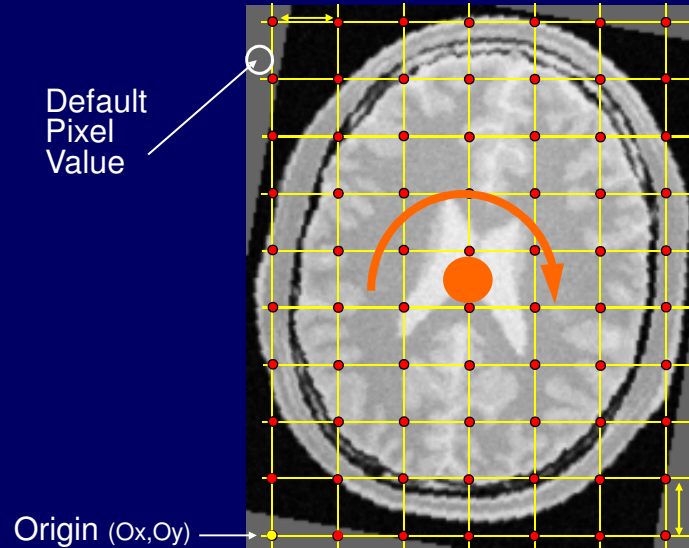


Image Registration

Lecture 7

51

Resample Image Filter

```
#include "itkImage.h"
#include "itkResampleImageFilter.h"
#include "itkCenteredRigid2DTransform.h"
#include "itkNearestNeighborInterpolateImageFunction.h"

typedef itk::Image< char, 2 > ImageType;

ImageType::Pointer inputImage = GetImageSomeHow();

typedef itk::ResampleImageFilter< ImageType > FilterType;

FilterType::Pointer resampler = FilterType::New();

ImageType::SizeType size;
size[0] = 200;
size[1] = 300;

ImageType::IndexType start;
start[0] = 0;
start[1] = 0;
```

Image Registration

Lecture 7

52

Resample Image Filter

```
ImageType::PointType origin;
origin[0] = 10.0; // millimeters
origin[1] = 25.5; // millimeters

ImageType::SpacingType spacing;
spacing[0] = 2.0; // millimeters
spacing[1] = 1.5; // millimeters

resampler->SetOutputSpacing( spacing );
resampler->SetOutputOrigin( origin );

resampler->SetSize( size );
resampler->SetOutputStartIndex( start );

resampler->SetDefaultPixelValue( 100 );

resampler->SetInput( inputImage );
```

Resample Image Filter

```
typedef itk::NearestNeighborInterpolateImageFunction<
    ImageType,
    double > InterpolatorType;

InterpolatorType::Pointer interpolator = InterpolatorType::New();

resampler->SetInterpolator( interpolator );

typedef itk::CenteredRigid2DTransform< double, 2 > TransformType;

TransformType::Pointer transform = TransformType::New();

resampler->SetTransform( transform );

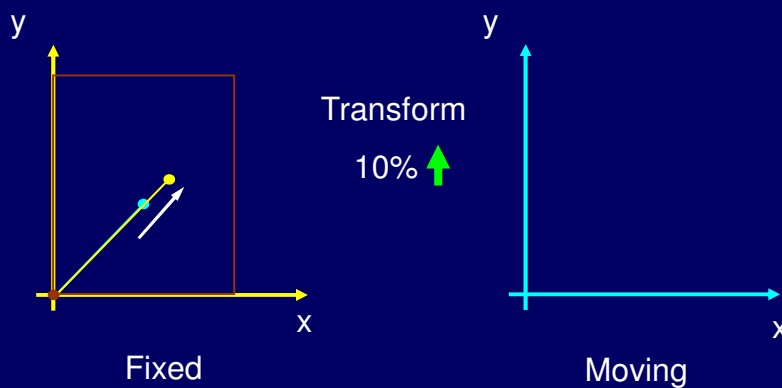
TransformType::InputPointType center;

center[0] = origin[0] + spacing[0] * size[0] / 2;
center[1] = origin[1] + spacing[1] * size[1] / 2;
```

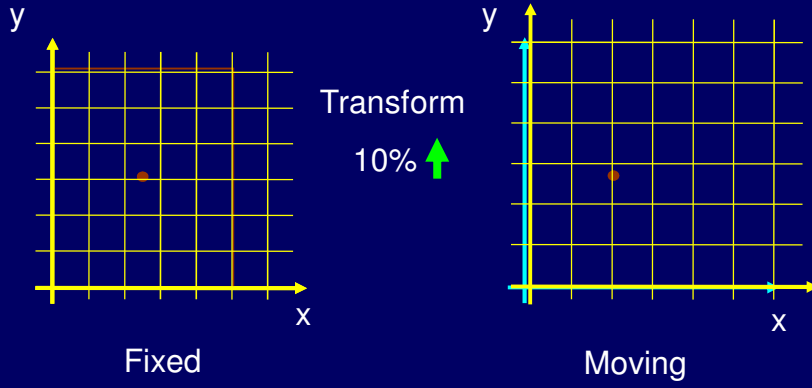
Resample Image Filter

```
transform->SetCenter( center );  
angle = 10.0 * 3.1514 / 180.0;  
transform->SetRotation( angle );  
transform->ComputeOffset ();  
resampler->Update ();  
const ImageType * outputImage = resampler->GetOutput ();
```

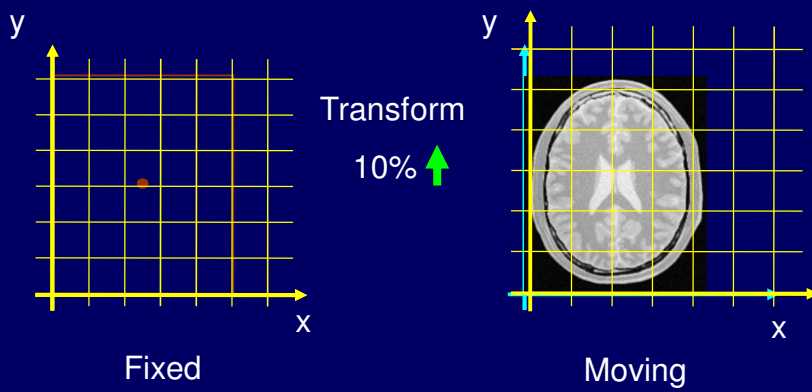
Scaling



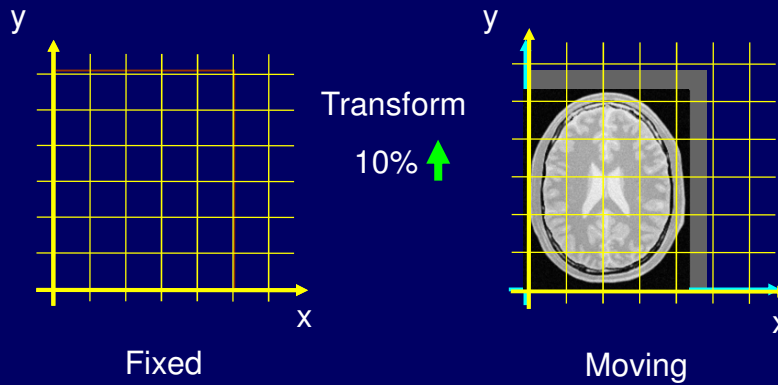
Scaling



Scaling



Scaling



Resample Image Filter

```
#include "itkImage.h"
#include "itkResampleImageFilter.h"
#include "itkScaleTransform.h"
#include "itkNearestNeighborInterpolateImageFunction.h"

typedef itk::Image< char, 2 > ImageType;

ImageType::ConstPointer inputImage = reader->GetOutput();

typedef itk::ResampleImageFilter< ImageType > FilterType;

FilterType::Pointer resampler = FilterType::New();

ImageType::RegionType region = inputImage->GetBufferedRegion();

resampler->SetSize( region->GetSize() );

resampler->SetOutputStartIndex( region->GetIndex() );
```

Resample Image Filter

```
resampler->SetOutputSpacing( inputImage->GetSpacing() );
resampler->SetOutputOrigin( inputImage->GetOrigin() );

typedef itk::ScaleTransform< double, 2 > TransformType;

TransformType::Pointer transform = TransformType::New();

TransformType::ScaleType scale;
scale[0] = 1.1;
scale[1] = 1.1;
transform->SetScale( scale );

resampler->SetTransform( transform );

resampler->SetDefaultPixelValue( 100 );

resampler->SetInput( inputImage );

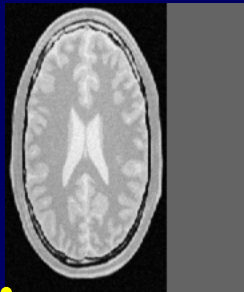
writer->SetInput( resampler->GetOutput() );
writer->Update();
```

Quiz #5

Fixed Image

Size: 181x217 pixels

Spacing: 0.8 x 0.9 mm



Origin: (0,0)



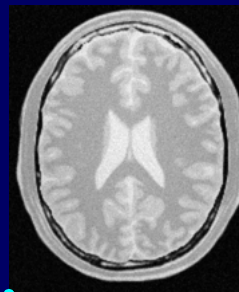
Transform ?

1. Scale X
2. Scale Y
- a) 1.50
- b) 0.75

Moving Image

Size: 181x217 pixels

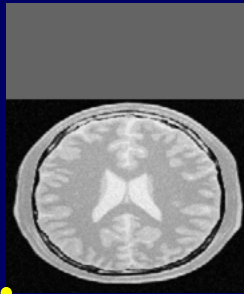
Spacing: 0.8 x 0.9 mm



Origin: (0,0)

Quiz #6

Fixed Image
Size: 181x217 pixels
Spacing: 0.8 x 0.9 mm



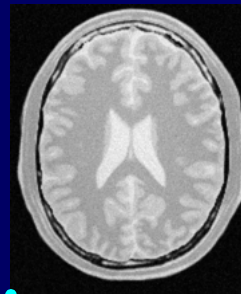
Origin: (0,0)



Transform ?

1. Scale X
 2. Scale Y
- a) 1.50
b) 0.75

Moving Image
Size: 181x217 pixels
Spacing: 0.8 x 0.9 mm



Origin: (0,0)